We called it RAID because it kills bugs dead

devblogs.microsoft.com/oldnewthing/20200317-00

March 17, 2020

Raymond Chen

The history of defect tracking in the Windows team goes back to Windows 1.0, which used a text file.

After Windows 1.01 released, a bunch of people in the apps division got together and threw together a bug tracking database. Because hey, a database, wouldn't that be neat?

The name was chosen by vote among the team, and the selected name was RAID, which is the name of a brand of insecticide whose advertisements in the United States use the tag line "Kills bugs dead." The icon for the program was a can of bug spray, naturally.

The letters RAID were retroactively declared to be an acronym for "Reporting and Incidents Database", but nobody knew that or cared. It was RAID.

After you built a bug query, you could save it for future use, and the file extension was `.rdq`, short for "RAID Query".

The name RAID was linguistically productive, because you can "RAID a bug", which means "File a bug in the project's RAID database." The `.rdq` also could be used as its own noun, meaning the query file. "Can you send me the .rdq for the bugs we are reviewing tomorrow?"

The database was written back in the days of 16-bit computing, so naturally it had a limit of 32,767 bugs. This was sufficient for many years, but eventually products encountered the record limit and had to "roll over" to new databases, where all bugs from the old database that hadn't yet been closed were copied to a new database (and received new record numbers), and the old database was put into read-only mode.

Naturally, this created confusion when you were reading through some code, and it had a comment like "This fixes bug 3141," with no indication as to which bug database that bug number refers to.

I think Windows 95 went through three RAID databases during its life.

The original authors of RAID had no idea that their little bug tracking database tool would be the primary defect tracking tool across all of Microsoft for multiple decades. If they had known, they might have been too scared to write it. When looking back on the origin of RAID, one of the original developers confessed, “It really wasn’t made to last that long. Sorry!”

Another scalability problem was that by the time the Windows XP project was chugging along, you would get into situations where there were so many people using RAID at once that the server would simply stop accepting new connections. When the ship room convened to go over the state of the Windows project, they sometimes had to call into operations and ask them to kill a few active connections to the back-end database so that the ship room could connect.

It was clear that RAID was being pushed far beyond what it was originally designed. A new defect tracking system was developed, named *Product Studio*, because naming your app *Something Studio* was fashionable at the time.

Product Studio didn’t have a limit of 32,767 records. It used a three-tier architecture for improved reliability and flexibility. It supported file attachments!

Product Studio served as the primary bug-tracking database for many years. But even with its improved architecture, you often ran into cases where the app stopped responding and simply told you “There was an error contacting the middle tier.”

I liked to joke that we should just get rid of that middle tier. It’s always the one that’s causing problems.

Product Studio kept things going until Windows 8, at which point Windows switched to on-premise Team Foundation Services for work item tracking.

The most recent move was in Windows 10, when the Windows team switched to *Visual Studio Online* for its work item tracking database. Mind you, that doesn’t mean that things have been stable, because the name of the service changed from *Visual Studio Online* to *Visual Studio Team Services*, and then again to *Azure DevOps Services*.

Even Azure DevOps wasn’t big enough to contain all of the Windows work items. Periodically, old work items are archived and moved to another project.[1] But at least the remaining work items didn’t get renumbered. They kept their old numbers, thank goodness.

[1] Unfortunately, the archive project renumbers the work items. Fortunately, the original work item is remembered in the title, so you can do a search for *originalid:3141* to find the old work item known as number 3141.

Raymond Chen

Follow